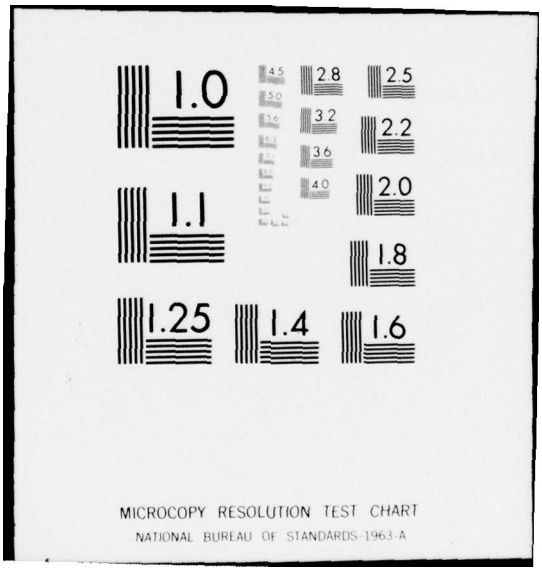
AD-A036 491 DARTMOUTH COLL HANOVER N H DEPT OF MATHEMATICS F/G 5/7
NATURAL LANGUAGE DATA BASE QUERY: USING THE DATA BASE ITSELF AS--ETC(U)
FEB 77 L R HARRIS N00014-75-C-0514
UNCLASSIFIED TR77-2 NL

1 OF 1
AD A036491

END
DATE FILMED
3-77


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

NATURAL LANGUAGE DATA BASE QUERY:

Using the data base itself as the definition of world knowledge and as an extension of the dictionary

Technical Report TR 77-2

Larry R. Harris

February 1977

Research partially sponsored by Office of Naval Research
Contract ONR N0014-75-C-0514

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R&D
(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)
DARTMOUTH COLLEGE
HANOVER, NEW HAMPSHIRE 03755

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b GROUP

3. REPORT TITLE
NATURAL LANGUAGE DATA BASE QUERY: Using the data base itself as the definition of world knowledge and as an extension of the dictionary.

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
SCIENTIFIC TECHNICAL REPORT, Jan 1976 — Feb 1977

5. AUTHOR(S) (Last name, first name, initial)
HARRIS, Larry R. / Harris

6. REPORT DATE
Feb. 1977

7a. TOTAL NO. OF PAGES
23

7b. NO. OF REFS
2

8a. CONTRACT OR GRANT NO.
N00014-75-C-0514 New

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
Dartmouth College Mathematics Dept. Technical Report TR 77-2

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. AVAILABILITY/LIMITATION NOTICES
Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce for sale to the general public.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY
Office of Naval Research
Code 437
Arlington, VA

13. ABSTRACT

This paper raises two issues that heretofore have not been dealt with in any previous natural language data base query system. These issues arise because of the everpresent need for world knowledge in the understanding of English, and also becuase of the particular way in which information is stored in a data base. The solutions to these problems described in this paper require only existing state of the art data base technology.

DD FORM 1 JAN 64 1473

UNCLASSIFIED
Security Classification

404 325

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Artificial intelligence computational linguisties data base data base retrieval English language processing Man machine communication Natural language Natural language processing Parsing Query language Question-answering Semantics of Natural Language						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2*a*. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2*b*. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year, or month, year. If more than one date appears on the report, use date of publication.

7*a*. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7*b*. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8*a*. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8*b*, 8*c*, & 8*d*. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9*a*. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9*b*. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through _____."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users *shall request* through _____."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through _____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.

GPO 886-551

NATURAL LANGUAGE DATA BASE QUERY: Using the data base itself as the definition of world knowledge and as an extension of the dictionary

Larry R. Harris
Mathematics Department
Dartmouth College
Hanover, New Hampshire 03755

Abstract

This paper raises two issues that heretofore have not been dealt with in any previous natural language data base query system. These issues arise because of the everpresent need for world knowledge in the understanding of English, and also because of the particular way in which information is stored in a data base. The solutions to these problems described in this paper require only existing state of the art data base technology. Systems based on these ideas serve as an example of a currently attainable, yet usable, natural language query system. As such they serve as a counter example of the philosophy that natural language processing is an all or nothing situation.

I. Introduction

Natural language data base query has long been recognized as a useful application of AI techniques. The state of the art in both natural language processing and in data base management systems (DBMS) has already reached the point where the two could be married to provide a useful access medium for untrained users.

What has impeded the useful application of the most successful natural language systems such as Winograd[72] and Woods[72]. Most people agree that the performance level is high enough, as shown by the LUNAR system's success at the Geology conference. Why then, have the techniques not been successfully utilized?

Basically the answer reduces to economics. The cost of running any of these systems is too high. The cost of a computer big enough to run them is too high. Most important the startup cost of applying these programs to a new area of discourse is too high. These costs are only offset by the unsubstantiated claims of higher user efficiency in a natural language environment. As of yet, no one has chosen to pay the price.

Beyond these questions of cost effectiveness lurk other problems that hinder the successful application of past systems. Basically these problems are related to the size of today's existing data bases. Because natural language query systems such as Wood's and Petrick's, impose a partition between the understanding and the retrieval functions, they face potentially insurmountable problems when applied to large data bases.

Existing research in knowledge representation is an attempt to merge these two functions. That is, the data base and the internal structures used by the parser would be one and the same. We keenly await the results of this research. An alternate approach is to more closely couple the understanding function with the data base by utilizing the high performance data base technology that exists today.

The basic thesis of this paper is that it is wholly infeasible to design natural language query systems that do not make use of the data base itself as a definition of world knowledge and as an extension of the dictionary. In Section II we develop this argument fully. Section III presents a proposed solution, along with a brief discussion of why the solution is feasible in terms of existing DBMS technology. Section IV discusses the impact of this on the basic design of a real system.

II The Problem.

Assume for a moment that all of the problems involved with natural language understanding were solved by an extension of any of the current approaches. Consider what problems would be encountered in applying this newly developed system to the environment of data base query.

First, we must note that all the systems we have discussed make use of an auxilliary dictionary, that typically contains the root form of words, their syntatic category and other useful bits of information such as special sufficies etc. All the existing natural language systems expect that all words that will appear in a sentence can be found, in one form or another, in this dictionary.

Now we can see some problems beginning to arise. Typical data bases involve thousands or even millions of English words. If we are forced to include all of this in our linguistic dictionary, then we would nearly duplicate the data base! Furthermore, we must realize that real data bases are rarely stagnant, they can change daily or in some cases continually. Updates to the data base would require corresponding changes to be made to the dictionary. Furthermore these changes in the dictionary are not always trivial to make, since they involve enumerating each word's syntactic category and all of its lexical

and word sense ambiguities. In the best cases this could be done by any competent computational linguist who had a working knowledge of the program. In the worst case actual programming changes may have to be made to use the new word correctly. In any case, it should be noted this is not a task that could easily be automated or performed by an ordinary programmer.

A common reaction to this dilemma is to solve it by entering the unabridged dictionary once and for all, feeling that this will solve the problem. Not true. Much of what the data base contains is a limited form of world knowledge. Often they are about people, places, and things. Thus, they often deal with proper names and composite names, hitting just at the unabridged dictionary's weakest point. You won't find much in the unabridged dictionary about proper names like "Albert Mahoney", or "South Podunk Falls". Also composite groups such as "skymist blue", or "executive secretary" won't be found. Yet all of these are very likely to appear in some data base, and thus very likely to appear in some query regarding that data base.

These last examples bring up another separate, but related problem. Since the natural language processor must at some point relate the query to the actual data base, some "meaning" of these words must also be given in the dictionary. For example. answering the questions "Who is in Debuque?" requires knowing that Debuque is a city and this may appear in the city field of the

data base. "How many skymist blue cars were sold in 1975?" requires knowing that "skymist blue" is a color as opposed to a manufacturer or a body type. It should be emphasized that a pure syntactic parse indicating that "skymist" modifies "blue" which modifies "car" is insufficient to formulate a search to the data base. We must somehow have access to the fact that "skymist blue" may appear in the color field.

This looks like an opportune time to claim that general world knowledge will solve the problem. After all, if having a list of all the cities in the world isn't world knowledge, what is? But that is exactly what would be required to solve the problem this way, a list of cities, a list of colors, a list of names, etc., etc.

If this is beginning to sound like another data base, you're wrong. It's beginning to sound like the same data base to which the queries are directed. The answer is clear. The data base itself must be used as both an extension of the dictionary and as a definition of world knowledge.

III A Solution.

Exactly what does it mean to say that the data base is the definition of world knowledge. It becomes more clear when you ask someone how they understand the queries "What cars are green?" and "What cars are Fords?" In one case you search the color field in the other you search the manufacturer field. But how did you know to do this? You called upon your world knowledge to identify "green" as a color and "Ford" as a manufacturer.

Many people jump to the conclusion that to use world knowledge to solve this means to have access to all the possible colors and manufacturers of cars. But this ignores the fact that people's world knowledge is not always complete. For example you might have trouble responding to "Which ones are taupe?", if you didn't know that "taupe" is indeed a color. Furthermore you probably bias the reading of "Are any of them Fords?" to thinking of cars when it makes perfect sense as a person's name.

To say that we use the data base as a definition of world knowledge is to say, for example, that if a word appears in the color field then it is a color, and if it does not appear in the color field then it is not a color. Similarly we define cities. states and names, in fact everything in the data base.

You may argue that green is a color whether or not any cars are colored green. This is of course true, but the same argument

holds for the color traube. We hope to obtain a reasonable level of competence with an incomplete definition of world knowledge, just as those of you who just learned that traube is a color have so aptly demonstrated is possible.

It might be thought that defining world knowledge in this way will make it impossible to respond to questions like "How many green cars are there?" when the word "green" does not appear in the data base. In fact this is not a problem, and all questions of this type can be answered without fear of misinterpretation, since the answer is clearly "none" no matter what "green" means.

However, it is possible to misinterpret a question like "Which of them are green?" when "green" does not appear in the color field but does appear in another field, such as the name field. Assuming that the user was asking about green colored cars, we would erroneously generate a query about people named Green. The earlier example about Ford illustrates that people are likely to make the same kind of error. By echoing back our interpretation of the query, as is done in the sample dialog, the user can see if any such misunderstanding takes place.

In the case where "green" appears in both the color and name fields, we simply ask the user which was meant, unless the syntax of the sentence gives no further clues. For example, "Which are green?" would require an interaction with the user, whereas "Which

are colored green?" or "Which are green in color?" would not.

We argued earlier that we must use the data base as an extension of the dictionary as well as a definition of world knowledge. We now discuss exactly what this means and how these two uses are distinct. By treating the data base as an extension of the dictionary we are saying that we would like to be able to perform the same operations on the data base as we do on the dictionary. Furthermore, we would like to extract the same information from the data base that we extract from the dictionary.

First, let us take up this issue of what operations we perform on the dictionary. Primarily here I am speaking of morphology, stripping words down to their root form. To understand the sentences "Who are the secretaries?' and "Who has a secretarial job?" requires the ability to figure out how these two forms of "secretary" appear in the actual data base. We must be able to do this by performing operations on the data base much like we would perform on the dictionary.

There is one further use of the dictionary that must be performed in the data base, that of forming composite groups. For example the proper noun "New York" is best thought of as one word that happens to have a blank in it. However, we need to ascertain this fact by looking in the data base or else we might parse it as "New" modifying "York". In a sentence like "Who is the New York

area manager?" we must determine how the composites are formed without the luxury of having any of the last four words in the dictionary. Furthermore the composites could be "New York" "area manager" or "New York area" "manager" depending on exactly how they were stored in the data base. In cases like this the data base itself is clearly the preferable place to dynamically extract such information, since it may vary with time.

In order for the data base to be an extension of the dictionary we must also be able to extract the same information from it that we can from the dictionary. For example, one item we expect from the dictionary is the syntactic category of a word. We could, of course, store such information in the data base along with the actual words. This goes along with the rather obvious strategy of making the dictionary an actual file under the control of the data base management system. However, this does not avoid the issue of how these facts are entered in the data base, particularly as the data base is dynamically updated. In any case this is a massive effort, as well as a significant perterbation of the existing data base. It would be far more practical to at least attempt to leave the data base intact and try to work around the problem. In this way we can hope to interface directly to an existing data base, without changing it in any way. Clearly this is a desirable goal if it can be achieved.

But how can we hope to parse a sentence without knowing the

syntactic category of every word in the sentence? This is a very unusual situation, one in which we do know the semantic use of the word, namely how and where it appears in the data base, but not its syntactic use. If we could only use a parser that was forgiving enough to allow the use of such words in a sentence. building only the syntactic structure for what it recognized, then later when high level semantic analysis begins, we might be able to merge the semantic knowledge about these words into the overall semantic structure of the sentence. For example the sentence "Print the names and phone numbers of all of the secretaries" might generate the following incomplete semantic structure.

```
(FILE (EMPLOYEE))
(PRINT (NAME PHONE))
(SEARCH (UNKNOWN-FIELD = SECRETARIES))
```

This could be merged with the semantic knowledge extracted from the data base that "SECRETARY" actually appears in the JOB field. This would form the following complete semantic structure suitable for initiating a full query to the data base.

```
(FILE (EMPLOYEE))
(PRINT (NAME PHONE))
(SEARCH (JOB = SECRETARY))
```

Is it feasible to drive a commercial DBMS in this way? Can we afford to dynamically find all the fields a given word appears in? It turns out that depending on the design philosophy of the

DBMS it may be quite feasible to perform these operations. The sample dialogue that follows demonstrates that this approach falls well within the state of the art of DBMS.

Very often people conjure up images of sequential passes over the file to see if "green" appears in any of the fields. This, of course, would be totally out of the question. Access to the data is achieved in a number of different ways in all of the major DBMSs. Thus, the data base designer has the choice of using hash code techniques, data inversion, network structures, or relational structures. Of these, the mechanism most suited for the type of access we require is data inversion. It turns out that the questions about the existence of a word in the data base can be answered by primitive operations on an inversion index thus they are extremely efficient. In this sense we are using the inversion index as an associative store.

Before specifying exactly what data inversion is, let me preface the discussion with the fact that the natural language analysis couldn't care less how the answers are obtained. It is certainly not dependent on inversion. Any of DBMS techniques that exist now or in the future, that can answer these kinds of questions in an acceptable time frame, are acceptable.

The best example of an inverted data base is the index at the end of a book. If it were fully inverted, every word used in the

book would would appear (only once) in the index along with a set of pointers (page numbers) to where the word appears. Given such a book and such an index, how hard is it to tell if the word "aardvark" appeared in the book? Quite easy, since the index is alphabetized. In fact, in order to to answer that question you could throw away the list of pointers since you don't need to know the page numbers it appears on. In fact, you could throw away the book itself, since the answer is wholly contained in the index.

To invert an entire data base, you simply treat each field as a new book and invert each field. Thus, you could poll the fields by asking if a given word is in the index of any field. In this way you find out whether the word appears at all, and if so what fields it appears in.

Conceptually you could take this collection of indices and treat it as another book and invert it, creating a higher level index, that for any word in the data base would give a list of pointers to the inversion tables in which that word appears. In such a case the answer to our question is merely a single search in an alphabetized list. To my knowledge none of the commercial DBMS's maintain this higher level index structure.

These secondary indices are created for efficient searching and sorting of the records in the file. In fact, arbitrary search union and intersection as well as sorting can be defined as operations on the inversion tables themselves so that no data need

ever be retrieved from the file until it is known to satisfy the search criteria and be in the desired order. It is for this reason that the inverted indices are created and maintained by the DBMS as new data is entered. We are merely making use of an already existing structure within the DBMS, and making use of it in straightforward manner.

Tests performed on a 10 million byte data base indicate that response time is well under 5 seconds real time even when more than 20 calls of this type are made to the DBMS.

IV A Complete Methodology

In this section we discuss how a system might make use of a data base in this way. The basic distinction of this approach is that we will make several calls to the DBMS while trying to understand the sentence, as well as the one final call to actually retrieve the answers. Most other query facilities try to limit themselves to this one final call.

Basically the order of the processing is as follows. The query is broken down into individual words, each of which is looked up in the dictionary, and if not there in the data base. Morphology is automatically performed during each of these searches. In the cases where individual words are not found, composite groups are formed as they appear in the data base.

At this point syntactic analysis begins, potentially building several incomplete interpretations. The holes within these semantic structures can now be merged with knowledge gained from asking the data base about individual words, as illustrated earlier. Only one task remains, namely selecting the one interpretation that was intended by the user from the set of well formed interpretations that still remains. Once again we turn to the data base to aid in the resolution of this problem.

As an example of this situation consider the following sentence.

"TELL ME ABOUT GREEN FORD CARS."

Assume that we have generated interpretations based on the fact that "GREEN" appears in both the color and name fields, and "FORD" appears in the name and manufacturer fields. Thus the four search terms would be

1. (AND (COLOR = GREEN) (MANUFACTURER = FORD))
2. (AND (COLOR = GREEN) (NAME = FORD))
3. (AND (NAME = GREEN) (MANUFACTURER = FORD))
4. (AND (NAME=GREEN) (NAME=FORD))

It should be pointed out that if the user had chosen a richer syntactic expression of his query, most of these interpretations would not have been generated. However, let us take the example exactly as given.

How can we use the data base to help select the intended interpretation? We simply push the idea of using the data base as a definition of world knowledge a little further. By applying all four search expressions to the data base we can get a reading on how meaningful each query is, given the current state of the data base. We should clearly make note of the fact that this is not in any sense a reading on how likely it is that the user intended this interpretation. But by determining whether or not

each search term has zero, or positive hits in the data base we can employ the following very useful heuristic. If there is exactly one interpretation with positive hits we select that interpretation as the one intended by the user, with an appropriate echo of the interpretation to act as a warning. If there are several positive hit interpretations we must ask the user which of these he intended, as the heuristic is of no help in these cases. Finally if there are only zero hit interpretations we can safely answer "none" or "no" appropriately.

Thus, for our example, assuming that only interpretaion number one had positive hits, then this interpretations would be selected, and the echo that we were searching for green colored cars made by Ford, would be printed.

This heuristic, which may seem rather tenuous at first, is based on the premise that people will tend to ask questions about things that are in the data base. In terms of the data base defining world knowledge, we are ascertaining which of the interpretations do not make sense with respect to the known state of the world. This heuristic is yet to fail, to my knowledge, in any real user session. It is however, very easy to conjure up situations in which it would fail, which is exactly why it is labelled a heuristic. The cost of such failure is small indeed considering that the user is warned of the

misinterpretation and can always rephrase the sentence using more syntactic clues to indicate the desired meaning.

The notion of asking the DBMS whether there are any hits on a given interpretation is not particularly expensive. People often imagine that each search requires making a pass over the data base retreiving each record to see if it meets the search criteria. However, by making use of the inversion tables the DBMS can perform the search logic on the index and immediately tell whether any records satisfy it or not. Thus the question can be answered without retreiving a single record, meaning that its basically an in core operation and thus quite fast.

V. Sample dialog

The techniques discussed in this paper are sufficient to process a large subset of the questions user is likely to ask. To best exemplify this level of competence, the following list of questions are given, all of which can be processed using the techniques described herein.

The questions pertain to a data base containing information about employees and cars. The initial questions show the use of the data base as a definition of world knowledge ,since neither the words "Ford" nor "green" appear in the dictionary. The interpretation of these questions is therefore dependent on the contents of the data base, ie, whether 'green" is in the name or the color field, or both. The next questions illustrate the use of the data base as an extension of the dictionary, since the phrases, "Vice President" and "Los Angeles' do not exist in the dictionary and therefore, must be pieced together. The remaining sentences illustrate the overall level of linguistic complexity that can currently be dealt with. Included are examples of the use of pronouns and sentences fragments.

WHATS IN THE EMPLOYEE FILE?

WHAT FIELDS ARE IN THE FILE OF CARS?

WHICH CARS ARE FORDS?

WHICH OF THOSE ARE GREEN?

PRINT A MILEAGE REPORT BY MANUFACTURER FOR '71 VOLVOS AND PORSHES
INCLUDING THEIR MODEL AND COLOR.

LIST THE PHONE NUMBERS OF THE SINGLE WOMEN IN ST.LOUIS.

GIVE ME A SALARY HISTOGRAM FOR THEM.

GIVE ME A SORTED LIST OF NAMES OF ALL THE VICE PRESIDENTS
IN CHICAGO OR LOS ANGELES.

ARE THERE ANY PEOPLE WORKING AS SECRETARIES THAT EARN A SALARY
OF $5,000 OR MORE?

BROKEN DOWN BY MANUFACTURER, PRINT A LIST OF ALL THE '70 GREEN CARS
WITH OVER 50,000 MILES ON THEM.

FIND THE CARS MADE BY PORSCHE AND MADE IN'71.

BROKEN DOWN JOB, REPORT ON THEIR NAME, SALARY, AND PHONE.

SALARY OF EMPLOYEES EARNING > $40,000.

References

Winograd[72], T., Understanding Natural Language, Academic Press, 1972

Woods[72], W.A., et al, "The Lunar Sciences Natural Language Information System", Bolt Beranek and Newman Report 2378, June 1972